

Claims

1. Method for generating a user interface of a network node, whereas an application is structured into a core application part responsible for handling data objects and a viewer/controller application part responsible for displaying
5 said data and initiating actions on said data, wherein said viewer/controller application part is formed by said user interface, characterized

in that a screen mask creating module (240) for creating dynamically a screen mask of said user interface retrieves screen mask configuration data (320) and widget configuration data (310) over a network which are stored on a
10 central processing unit,

in that a widget creating module (230) generates at least one user interface component (10 - 18; 410) on the basis of one or more component patterns (411, 412) of the widget configuration data (310) and stores the at least one user interface component (10 - 18; 410) by means of a widget cache (210),

15 in that a screen mask of said user interface is generated by said screen mask creating module (240), wherein said screen mask comprises at least one component which is a component out of said components comprising the widget configuration data, and

in that said at least one component of said created screen mask is
20 assigned to at least one data object and/or dynamic of said components assigned to said screen mask based upon a user action on a user interface component and/or a data object.

2. Method according to claim 1, characterized in that said screen mask of said user interface is dynamically generated by said screen mask
25 creating module (240).

3. Method according to one of the claims 1 or 2, characterized in that said screen mask configuration data and/or widget configuration data are at least partially changed dynamically based upon one or more user actions.

4. Method according to one of the claims 1 to 3, characterized in that
30 said screen mask configuration data (320) is retrieved (S220), said screen mask configuration data (320) is parsed (S220) to obtain type information about said at least one component (10 - 18; 410) and to obtain individual settings of said at

least one component (10 - 18; 410), said at least one component (10 - 18; 410) is created (S230) by obtaining (S231) said at least one component (10 - 18; 410) on the basis of at least one component pattern (411, 412) corresponding to said type information and said individual settings is applied (S232) onto said at least one component (10 - 18; 410) and including said at least one component (10 - 18; 410) into said screen mask.

5. Method according to claim 4, characterized in that said at least one component (10 - 18; 410) are requested from a component pattern repository (210), which caches at least one component pattern (411, 412), said at least one component pattern (411, 412) are identified corresponding to said type information, and said at least one component (10 - 18; 410) are derived from said at least one identified component pattern (411, 412).

6. Method according to claim 5, characterized in that said component pattern repository (210) is initialized by retrieving (S110) said widget configuration data (310), which comprises widget configuration data about at least one component pattern (411, 412), parsing (S120) said widget configuration data (310), creating (S130) said at least one component pattern, and storing said at least one component pattern in said component pattern repository (210).

7. Method according to one of the claims 5 or claim 6, characterized in that said component pattern repository (210) contains statically said at least one component pattern (411, 412) during runtime of said application.

8. Method according to one of the claims 1 to 7, characterized in that said at least one component (10 - 18; 410) are obtained by requesting said at least one component (10 - 18; 410), retrieving (S110) a widget configuration data (310), which comprises widget configuration data about at least one component pattern (411, 412), identifying said widget configuration data about said at least one component pattern (411, 412) corresponding to said extracted type information, parsing (S120) said identified widget configuration information, creating (S130) said at least one component pattern, deriving said at least one component (10 - 18; 410) from said at least one component pattern (411, 412).

9. Method according to one of the claims 1 to 8, characterized in that said at least one component (10 - 18; 410) are obtained by cloning said at least

one component pattern (411, 412) to obtain said at least one component (10 - 18; 410).

10. Method according to one of the claims 1 to 9, characterized in that said widget configuration data (310) comprises default widget configuration
5 information about said at least one component pattern (411, 412) such that components (10 - 18; 410) obtained from said least one component pattern (411, 412) have default settings valid for substantially all components used in the user interface.

11. Method according to one of the claims 1 to 10, characterized in
10 that said screen mask configuration data (320) comprises screen mask configuration data about at least one component (10 - 18; 410) to adapt said component (10 - 18; 410), which is obtained from said corresponding component pattern (411, 412), to requirements presupposed by said screen mask to be created.

12. Method according to one of the claims 1 to 11, characterized in
15 that said screen mask configuration data (320) is an XML-encoded screen mask configuration data, which is based on a screen mask document type description (DTD; 340).

13. according to one of the claims 1 to 12, characterized in that said
20 widget configuration (310) is an XML-encoded widget configuration, which is based on a widget document type description (DTD; 330).

14. Software tool for establishing a user interface (GUI), comprising
program portions for carrying out the operations of any one of the claims 1 to 13,
when said program is implemented in a computer program for being executed on
25 a microprocessor-based device, processing device, a terminal device or a network device.

15. Computer program product for establishing a user interface (GUI),
comprising loadable program code sections for carrying out the operations of
any one of the claims 1 to 13, when said program code is executed on a
30 microprocessor-based device, processing device, a terminal device or a network device.

16. Computer program product for establishing a user interface, wherein said computer program product is comprising program code sections stored on a computer readable medium for carrying out the method of any one of the claims 1 to 13, when said computer program product is executed on a
5 microprocessor-based device, processing device, a terminal device or a network device.

17. Terminal device adapted to establish a user interface, which is operable by a user to operate an application executed by said terminal device. which comprises a screen mask creating component (240) for creating
10 dynamically a screen mask of said user interface (GUI), comprising:

a retrieval component (260, 270) for retrieving a screen mask configuration (320), which comprises screen mask configuration data about at least one component (10 - 18; 410),

a parsing component (250, 240) for parsing said screen mask
15 configuration (320) to obtain type information about said at least one component (10 - 18; 410) and to obtain individual settings of said at least one component (10 - 18; 410),

a widget creating component (240) for obtaining said at least one component (10 - 18; 410) on the basis of at least one component pattern (411, 412) corresponding to said type information and for applying said individual
20 settings onto said at least one component (10 - 18; 410), and

a linking component (430) for linking said at least one component (10 - 18; 410) to at least one data object (460, 465).

18. Terminal device according to claim 17, comprising a component
25 pattern repository (210) which caches at least one component pattern (411, 412) and from which at least one component (10 - 18; 410) is requested and an identification component (240) for identifying at least one component pattern (411, 412) corresponding to said extracted type information, wherein said widget creating component (240) is adapted to derive at least one component (10 - 18;
30 410) from said at least one identified component pattern (411, 412).

19. Terminal device according to claim 17, wherein said terminal device comprises as further components for initialization of said component pattern repository (210) a retrieval component (260, 270) for retrieving a

component configuration (310), which comprises component configuration data about at least one component pattern (411, 412), and a parsing component (250, 230) for parsing said component configuration information, wherein said widget creating component (230) is adapted to create said at least one
5 component pattern (411, 412) and to store said at least one created component pattern (411, 412) in said component pattern repository (210).

20. Terminal device according to claim 19, comprising a retrieval component (260, 270) for retrieving a component configuration (310), which comprises component configuration information about at least one component
10 pattern (411, 412), an identification component (240) for identifying said component configuration information about said at least one component pattern (411, 412) corresponding to said extracted type information, and a parsing component (250, 230) for parsing said identified component configuration information, wherein said widget creating component (230) is adapted to create
15 said at least one component pattern (411, 412) and to derive said at least one component (10 - 18, 510) from said at least one component pattern (411, 412).